Whitepaper

Remote Password Changing

For Managed Services clients

Last Updated 05/26/2021

Contents

Summary

Remote Password Changing (RPC) benefits operational efficiency and security by providing:

Ease of account administration using shared client-side credentials

Individual account Multi-Factor Authentication (MFA) before accessing shared credentials

Automatic password change after use

Complete audit log of individual and shared credential usage

RPC behavior

Activity	Frequency	Initiated by
Heartbeat (validity) check	24 hours	System
Automatic password change	30 days (typical)	System at expiration
Checkout/view password	As needed	Operations
Check-in password change	As needed	Operations or system at checkout interval expiration

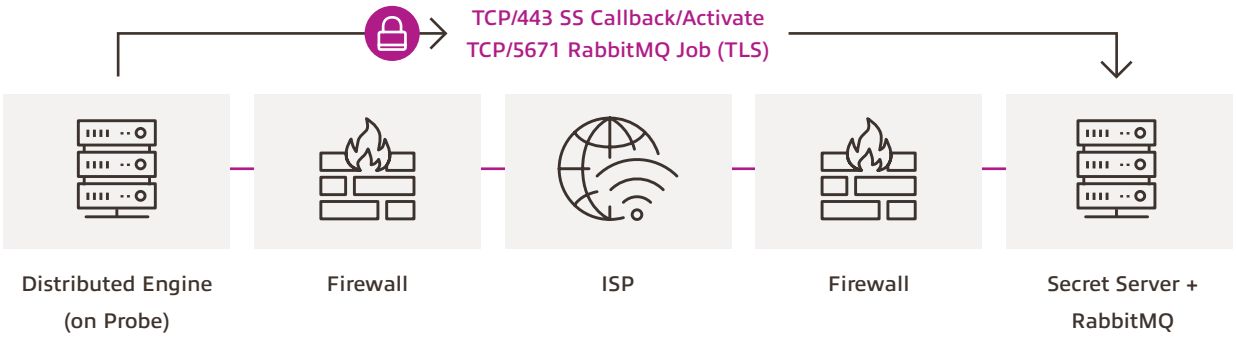
The system will complete the check-in procedure and change the password when a password is checked in or when the checkout interval has expired.

Transport security

The Distributed Engine uses standards-based TLSv1.2 for two primary connections:

- Registration of Distributed Engine to Secret Server®
- Job polling via RabbitMQ® connector

Any edge client firewalls must allow TCP/443 and TCP/5671 outbound to central Secret Server for these connections.



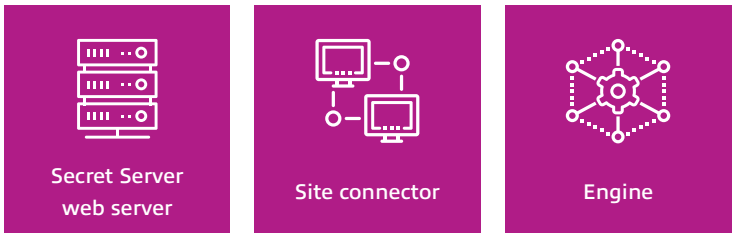
Ports required from DE to client system(s)

Depending on which system type is in scope for RPC, Distributed Engine must be allowed TCP access to internal systems via client firewalls according to the following changer types:

Active Directory®	Port 369 or 636

Components of Distributed Engine

Secret Server Distributed Engine has three installed components:



The web server generates messages and places them in the site connector. The Engine connects to the site connector and retrieves messages, processes them, and then hands the results back to Secret Server. Secret Server organizes network locations through sites. A site is a virtual container for secrets and discovery sources that belong within a specific client network location. Each secret and discovery source has an assigned site. Each Engine must be assigned to only one site.

Engine authentication to Secret Server

When an engine starts, it authenticates as follows:

Engine <-> Secret Server handshake

1. Engine calls Secret Server — Secret Server generates a public/private 4096-bit RSA key pair (server public key and server private key) and returns the public key (server public key).

2. Engine creates an authentication request, which contains the identifying information for the engine.

3. Engine generates its own public/private 4096-bit RSA key pair (engine public key and engine private key).

4. Engine encrypts its authentication request with the server public key.

5. Engine submits the encrypted authentication request to Secret Server along with the engine public key.

6. Secret Server decrypts the authentication request using the server private key.

7. Secret Server processes the authentication request — creates an engine record if necessary, checks activated status, etc.

   a. If the engine is inactive or there's another authentication problem, an error message is returned.

   b. If there's no authentication problem, Secret Server pulls the appropriate per-engine AES 256 symmetric key (engine symmetric key) from the database, encrypts it with the engine public key, and sends it back to the engine.

8. When the engine receives the symmetric key message, it decrypts it using the engine private key and retrieves the engine symmetric key. From now on, all communication between the engine and Secret Server will use the engine symmetric key.

Configuration request

1. Engine sends message to Secret Server asking for its configuration information.

2. Secret Server responds with the site, site connector connection information, site connector credentials, and the site symmetric key (an AES 256 symmetric key used to encrypt/decrypt the messages for the site) for the engine.

Connecting to the site connector and message processing

1. Using the site connector connection information, the engine connects to the site connector and starts listening for messages for its site.

2. When the engine has capacity to process messages and messages are available, the engine retrieves the message, decrypts it using the site symmetric key, processes the message, and sends the result back to Secret Server.

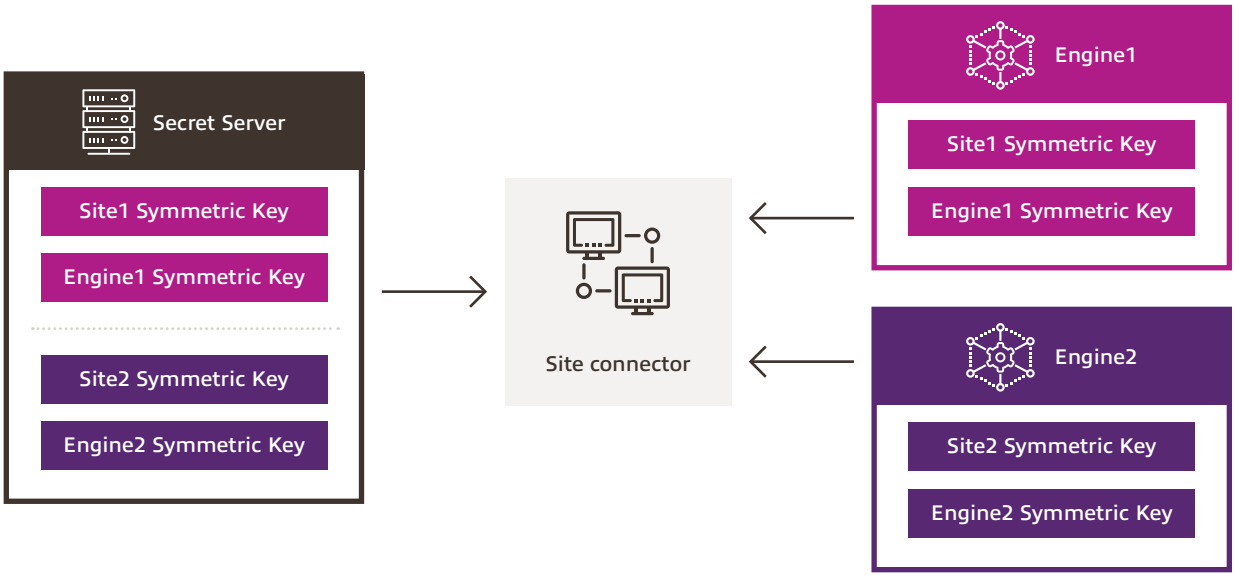
Encryption and sites

Each site has its own AES 256 symmetric key (site symmetric key) that's used to encrypt and decrypt messages for that particular site. As a result, each engine has access to at most one site symmetric key. An example flow for a Secret Heartbeat is below.

1. Secret Server loads the encrypted secret from the database.
2. Secret Server decrypts the secret using the secret's AES 256-bit key.
3. Secret Server creates a Secret Heartbeat message for the decrypted secret.
4. Secret Server encrypts the message with the site symmetric key.
5. Secret Server sends the message to the site connector.
6. An engine assigned to the specific site connects to the site connector and retrieves the message.
7. The engine decrypts the Secret Heartbeat message using the site symmetric key.
8. The engine processes the Secret Heartbeat message and creates a response object.
9. The response object is encrypted using the engine symmetric key and submitted to Secret Server.

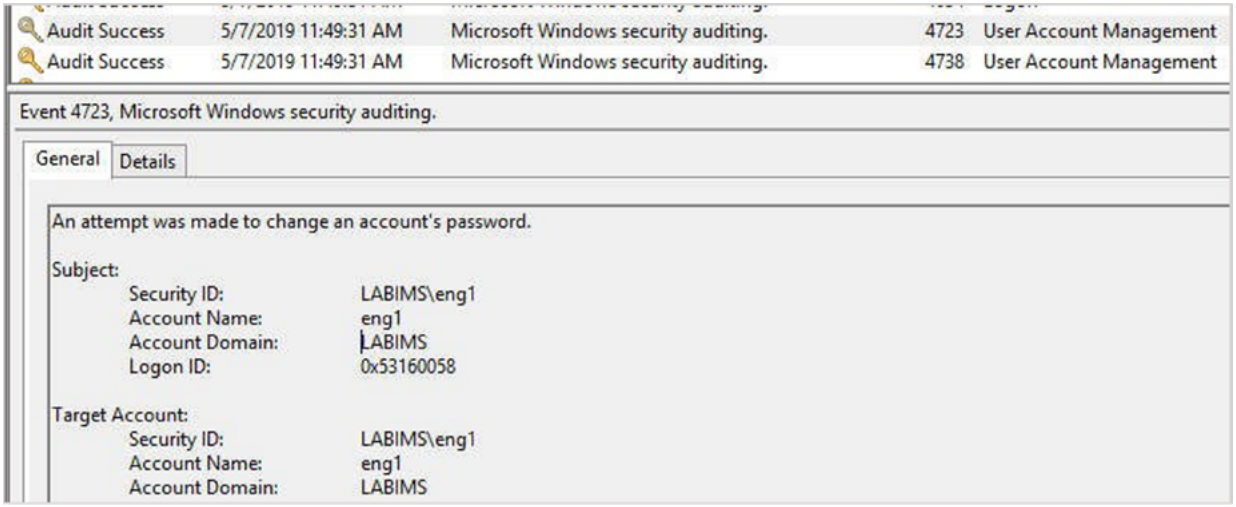
Site symmetric key locations

The following diagram illustrates a scenario in which Secret Server is configured with two sites (Site1 and Site2), with one engine each (Engine1 and Engine2). Note that each engine has access only to the keys it needs — namely, its own engine symmetric key (for communicating with Secret Server) and the site symmetric key for the site it's assigned to. Engine1 doesn't have access to the engine symmetric key for Engine2, nor does it have access to the site symmetric key for Site2. As shown below, the site connector doesn't have access to any of the keys, and thus can't decrypt any of the messages it holds.



Audit and logging

Windows Event Viewer



Individual to shared secret mapping

Date Recorded	User	Folder Path	Secret Name	Action	
5/7/2019 06:07 PM	ThycoticSystem	\Managed Services Clients\AAAA-Dist Engine Test	eng1	SECRET CHECKED IN	Password Changed
5/7/2019 06:07 PM	ThycoticSystem	\Managed Services Clients\AAAA-Dist Engine Test	eng1	CHANGE PASSWORD	Fields: (Password)
5/7/2019 06:06 PM	ThycoticSystem	\Managed Services Clients\AAAA-Dist Engine Test	eng1	SECRET SET FOR CHECK IN	
5/7/2019 05:51 PM	nsitmsp.com	\Managed Services Clients\AAAA-Dist Engine Test	eng1	PASSWORD DISPLAYED	
5/7/2019 05:51 PM	nsitmsp.com	\Managed Services Clients\AAAA-Dist Engine Test	eng1	VIEW	
5/7/2019 05:51 PM	nsitmsp.com	\Managed Services Clients\AAAA-Dist Engine Test	eng1	SECRET CHECKED OUT	